.S

Introduction to software development tools

make

.h

.c++

.c

Michael Opdenacker

Free Electrons

http://free-electrons.com

Created with OpenOffice.org 2.0

Thanks to Nicolas Rougier (Copyright 2003, http://webloria.loria.fr/~rougier/) for the Tux image

© Copyright 2005, Michael Opdenacker

Creative Commons Attribution-ShareAlike 2.0 license

Rights to copy

creative commons
COMMONS DEED

Attribution – ShareAlike 2.0

You are free

- to copy, distribute, display, and perform the work
- to make derivative works
- to make commercial use of the work

Under the following conditions

Attribution. You must give the original author credit.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

License text: http://creativecommons.org/licenses/by-sa/2.0/legalcode

© Copyright 2005
Michael Opdenacker
michael@free-electrons.com

Document sources, updates and translations:
http://free-electrons.com/articles/swdev

Corrections, suggestions, contributions and translations are welcome!

Best viewed with...

This document is best viewed with a recent PDF reader or with OpenOffice.org itself!

- Take advantage of internal or external hyperlinks. So, don't hesitate to click on them!
- Find pages quickly thanks to automatic search
- Use thumbnails to navigate in the document in a quick way

If you're reading a paper or HTML copy, you should get your copy in PDF or OpenOffice.org format on http://free-electrons.com/articles/swdev!

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Contents

Source management

- Program build automation
- Static code checkers
- Memory checkers

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Embedded Linux development tools

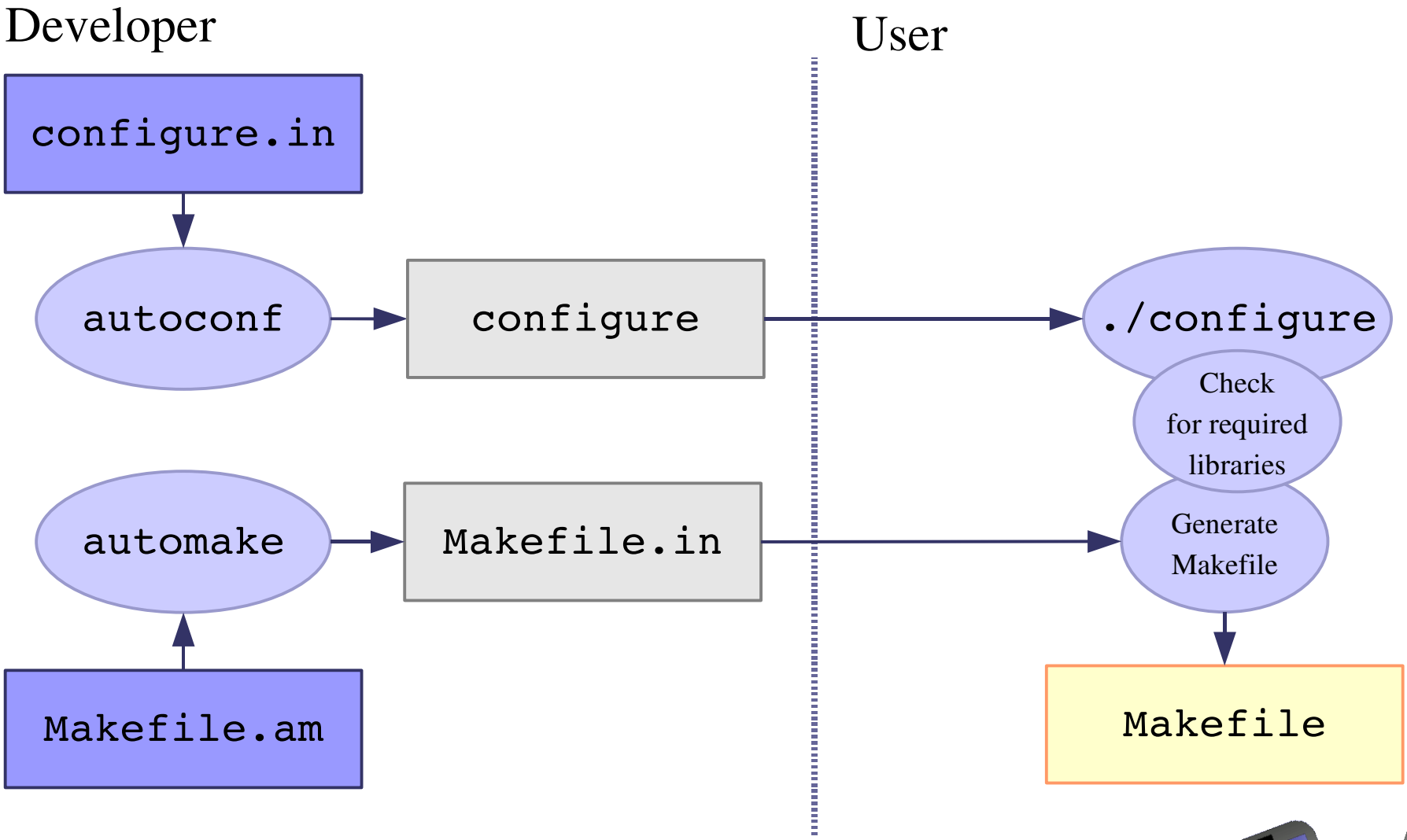
Source management

Program build automation

© Copyright 2005, Michael Opdenacker

Creative Commons Attribution-ShareAlike 2.0 license

GNU toolchain flow (simplified)



© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

GNU toolchain strengths and weaknesses

Universal solution for Unix systems

- Still used to build most Free Software C and C++ programs, in particular in GNU
- Very powerful and easy for end users:

```
./configure --prefix=/opt
make
make check
make install
```

- Causes of check failures on the user side sometimes not easy to figure out.
- No standard configure options
- Output very complex Makefiles

Difficult for developers to master

- Lots of details to learn and understand
- Tedious documentation
- Multiple input and output files
- Causes of check failures on the user side sometimes not very easy to figure out.

See http://freshmeat.net/articles/view/889/, http://freshmeat.net/articles/view/1702/ and http://freshmeat.net/articles/view/1715/ for discussions about the limitations of autoconf / automake / make and alternatives.

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

SCons (1)

http://www.scons.org/

- An alternative to `make`, written in Python
  A successor of the `Cons` program, written in Perl.
- Takes care of all the compiling and dependency handling job.
  Dependencies handled with MD5 checksums by default.
  Running `touch` on source files will have no effect!
- Also supports `autoconf` like capabilities: checking for existing libraries, headers or programs on the user system.
- Unlike `make`, `autoconf`, `automake`, no specific format!
  Access to all Python syntax and scripting facilities.

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

SCons (2)

- Native support for lots of compiler based languages:
  C, C++, D, Java, Fortran, Yacc, Lex, Qt and SWIG, TeX and LaTeX
  Easily extensible through user-defined builders.
- Supports building on multiple platforms:
  GNU/Linux, Unix, Windows NT, MacOS X, OS/2
- Built-in support for fetching source files from SCCS, RCS, CVS, BitKeeper and Perforce.
- ... and many more features!

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

SCons (3)

- Simple `SConstruct` file:

```
Library('foo', ['f1.c', 'f2.c', 'f3.c'])
Program('prog.c', LIBS=['foo', 'bar'], LIBPATH='.')
```

- Nice and friendly user manual full of examples:
  http://www.scons.org/doc/HTML/scons-user/book1.html

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Embedded Linux development tools

Source management

Static code checkers

© Copyright 2005, Michael Opdenacker

Creative Commons Attribution-ShareAlike 2.0 license

Splint

http://splint.org/, from the University of Virginia

- GPL tool for statically checking C programs for security vulnerabilities and coding mistakes
- Today's `lint` program for GNU/Linux.
  The successor of LClint.
- Very complete manual and documentation
- Doesn't support C++

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Debugging and analysis tools

Memory checkers

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

GNU Checker

http://gnu.org/software/checker/

GNU GPL tool to find memory errors at runtime

- Detects: reading uninitialized memory, writing in a freed zone, accesses outside of allocated memory, and using NULL pointers.
- Garbage collector: can display all memory leaks
- Works by replacing glibc's malloc library by its own
- Usage:

```
checkergcc -o example example.c
```

- Supports only: Intel GNU/Linux, Sparc Sun Solaris
- Latest release in 1998!

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

memcheck

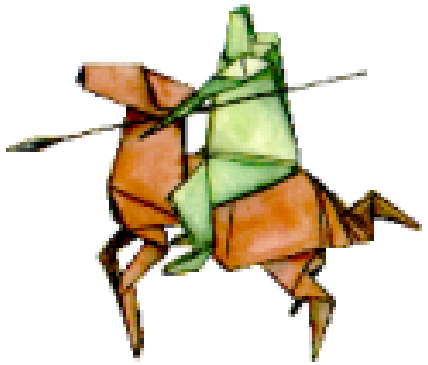
http://hald.dnsalias.net/projects/memcheck/

- GNU GPL tool for dynamic memory checking
- Works by replacing glibc's memory management functions by its own.
- Supports many GNU/Linux and other Unix architectures (unlike GNU Checker): `i386`, `arm`, `alpha`, `ia64`. Many others not tested.

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Valgrind (1)

http://valgrind.org/



- GNU GPL Software suite for debugging and profiling programs.
- Supported platforms: `x86`, `x86_64`, `ppc`
  Others: compile your program to these platforms to use Valgrind.
- Can detect many memory management and threading bugs.
- Profiler: provides information helpful to speed up your program and reduce its memory usage.
- The most popular tool for this usage.
  Even used by projects with hundreds of programmers

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Valgrind (2)

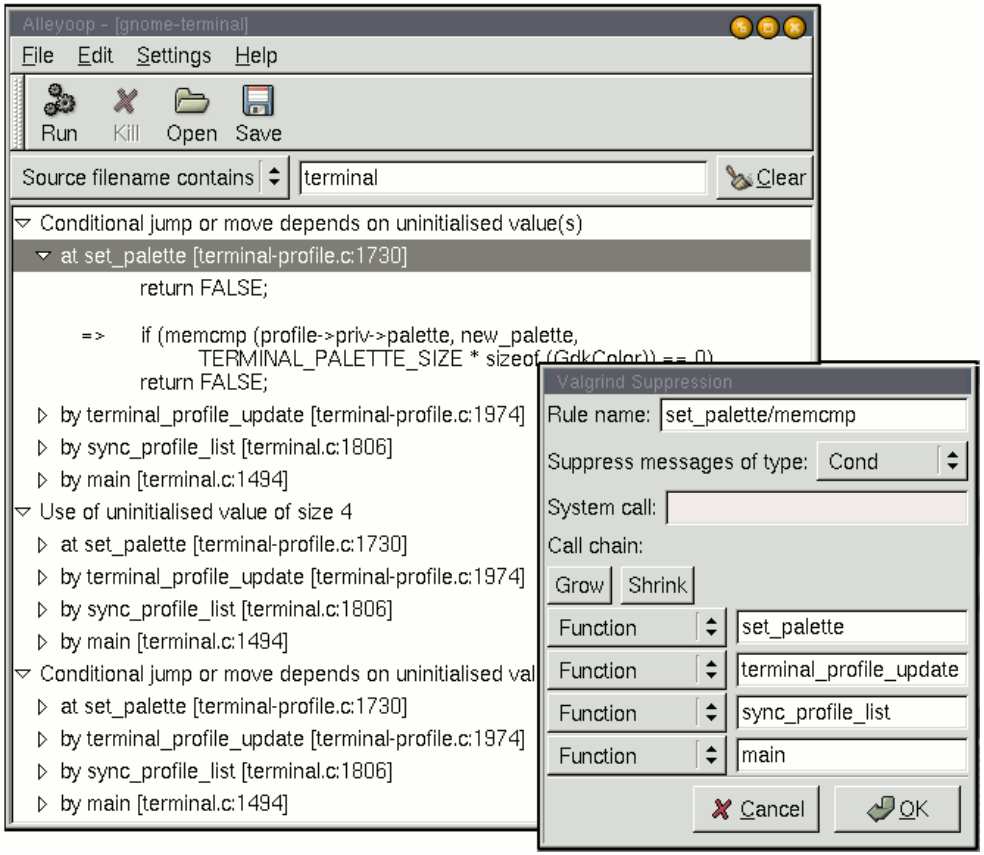
- Can be used to run any program, without the need to recompile it.
- Example usage
```
valgrind --leak-check=yes ls -la
```
- Works by adding its own instrumentation to your code and then running in on its own virtual x86 core. Significantly slows down execution, but still fine for testing!
- More details on http://valgrind.org/info/ and http://valgrind.org/docs/manual/coregrind_core.html#howworks

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Alleyoop

http://alleyoop.sourceforge.net/

Graphical front-end to Valgrind



© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

ElectricFence

`efence`, GNU GPL, from Bruce Perens
http://directory.fsf.org/devel/debug/ElectricFence.html

- Stops your program on the exact instruction that overruns or underruns a `malloc()` memory buffer.
- GDB will then display the source-code line that causes the bug.
- Works by using the virtual-memory hardware to create a red-zone at the border of each buffer - touch that, and your program stops.
- Supposed to work on any platform supported by Linux, whatever the cpu (provided virtual memory support is available)

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Related documents

This document belongs to the more than 1000 page materials of an embedded GNU / Linux training from Free Electrons, available under a free documentation license.

http://free-electrons.com/training

- Introduction to Unix and GNU/Linux
- Embedded Linux kernel and driver development
- Free Software tools for embedded Linux systems
- Audio in embedded Linux systems
- Multimedia in embedded Linux systems

http://free-electrons.com/articles

- Embedded Linux optimizations
- Embedded Linux from Scratch... in 40 min!
- Linux on TI OMAP processors
- Free Software development tools
- Introduction to uClinux
- Real-time in embedded Linux systems
- What's new in Linux 2.6?
- Java in embedded Linux systems
- How to port Linux on a new PDA

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

How to help

If you support this work, you can help ...

- By sending corrections, suggestions, contributions and translations
- By asking your organization to order training sessions performed by the author of these documents (see http://free-electrons.com/training)
- By speaking about it to your friends, colleagues and local Free Software community.
- By adding links to our on-line materials on your website, to increase their visibility in search engine results.

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license

Thanks

- To the OpenOffice.org project, for their presentation and word processor tools which satisfied all my needs.
- To the Handhelds.org community, for giving me so much help and so many opportunities to help.
- To the members of the whole Free Software and Open Source community, for sharing the best of themselves: their work, their knowledge, their friendship.

© Copyright 2005, Michael Opdenacker
Creative Commons Attribution-ShareAlike 2.0 license